LAMPIRAN

Lampiran 1 *Source code* MQTT di ESP8266

Code

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
const char* ssid = "Bersabar"; //deklarasi
const char* password = "77777777";
const char* mqtt_server = "192.168.43.229";
WiFiClient espClient;
PubSubClient client(espClient);
String inString = "";
int a = 0 ;
long lastMsg = 0;
char msg[50];
//char val = "";
int value = 0;
int count = 0;
String inChar;
char* d;
char* d1;
char* d2;
char* d3;
char* topik = "saifud/suhu";
char* topik1 = "saifud/pakan";
char* topik2 = "saifud/berat";
void setup()
{
       pinMode(BUILTIN_LED, OUTPUT);
       Serial.begin(115200);
       setup_wifi();
       client.setServer(mqtt_server, 1883);
       client.setCallback(callback);
}
void setup_wifi()
{
       delay(10);
       Serial.println();
       Serial.print("Connecting to ");
       Serial.println(ssid);
```

```
	WiFi.begin(ssid, password);
	while (WiFi.status() != WL_CONNECTED)
	{
		delay(500);
		Serial.print(".");
	}
	Serial.println("");
	Serial.println("WiFi connected");
	Serial.println("IP address: ");
	Serial.println(WiFi.localIP());
}
void callback(char* topic, byte* payload, unsigned int length)
{
	Serial.print("Message arrived [");
	Serial.print(topic);
	Serial.print("] ");
	for (int i = 0; i < length; i++)
	{
		Serial.print((char)payload[i]);
	}
	Serial.println();
	if ((char)payload[0] == '1')
	{
		digitalWrite(BUILTIN_LED, LOW);
	}
	else
	{
		digitalWrite(BUILTIN_LED, HIGH);
	}
}
void reconnect()
{
	while (!client.connected())
	{
		Serial.print("Attempting MQTT connection...");
		if (client.connect("ESP8266Client"))
		{
			Serial.println("connected");
			client.publish("outTopic", "hello world");
			client.subscribe("inTopic");
		}
```

```
            else
            {
                  Serial.print("failed, rc=");
                  Serial.print(client.state());
                  Serial.println(" try again in 5 seconds");
                  delay(5000);
            }
      }
}
String getValue(String data, char separator, int index)// memecah
data string berdasarkan sparator (, #)
{
      int found = 0;
      int strIndex[] = {0, -1};
      int maxIndex = data.length()-1;
      for(int i=0; i<=maxIndex && found<=index; i++)
      {
            if(data.charAt(i)==separator || i==maxIndex)
            {
                  found++;
                  strIndex[0] = strIndex[1]+1;
                  strIndex[1] = (i == maxIndex) ? i+1 : i;
            }
      }
      return    found>index    ?    data.substring(strIndex[0],
strIndex[1]) : "";
}
void loop()
{
      if (!client.connected())
      {
            reconnect();
      }
      client.loop();
      while (Serial.available())
      {
            inChar = Serial.readString();
            //inChar = "50#10/10/10 Rabu 45:45:45";
            String stmp = getValue(inChar,'#',0); //suhu
            String stmp1 = getValue(inChar,'#',1); //waktu
            String stmp2 = getValue(inChar,'#',2); //berat
            int tmp1= stmp1.toInt();
```

```
            String tb = "TERBUKA";
            String ttp = "TERTUTUP";
            d = (char *) stmp.c_str();
            d3 = (char *) stmp2.c_str();
            d1= (char *) tb.c_str();
            d2= (char *) ttp.c_str();
      client.publish(topik,d);
      client.publish(topik2,d3);
            if(tmp1>= 25 && tmp1 <= 30       {
                  if(true)
                  {
                        count = count + 1;
                        client.publish(topik1,d1);
      }
            }
            else
            {
                  client.publish(topik1,d2);
            }
      }
  }
```

Lampiran 2 *Source code* Arduino Uno

Code

```
#include "RTClib.h"
#include "HX711.h"
#include "RunningAverage.h"
#include <Servo.h>
#include <Wire.h>
#include <SoftwareSerial.h>
#include <OneWire.h>//untuk suhu
#include <DallasTemperature.h>
//suhu
#define ONE_WIRE_BUS 2  // Data wire is plugged into port 2 on the Arduino
DallasTemperature sensors(&oneWire);
RunningAverage myRA(10);
int suhu;
bool bukaP = false;
bool harian = true;
int hari = 10;
String pesan;
```

```
int count =0;
//servo
Servo servoPakan;
int pos = 0;
String d;
char* e;
int temp;
bool bukai = true;
bool bukau = true;
//RTC
String y;
String m;
String dat;
String dy = "";
String jam;
String ment;
String ertc;
int perubahan = 1;
String detk;
RTC_DS1307 rtc;
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"};
//loadcell
HX711 scale(7,6);
float calibration_factor = 150;
float units;
float ounces;
SoftwareSerial serial1 (11, 12); //Tx 12 Rx 11
void setup ()
{
       serial1.begin (115200);
       Wire.begin();
       Serial.begin(115200);
       servoPakan.attach(5);
       pesan += Serial.println("Bissmillah...");
       // Start up the library
       sensors.begin();
       while (!Serial); // for Leonardo/Micro/Zero
       if (! rtc.begin())
       {
              Serial.println("Couldn't find RTC");
```

```
                while (1);
        }
        if (! rtc.isrunning())
        {
                Serial.println("RTC is NOT running!");
                 rtc.adjust(DateTime(2017, 10, 30, 7, 55, 0));
                //rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
        }
        if (bukai == true)
        {
                bukaPakan();
                delay(3000);
                bukai = false;
        }
        if(bukai==false)
        {
                bukai = true;
                tutupPakan();
                //delay(2000);
                Serial.println("tutup pakan");
        }
        Serial.println("HX711 calibration sketch");
        Serial.println("Remove all weight from scale");
        Serial.println("After readings begin, place known weight on
scale");
        Serial.println("Press + or a to increase calibration factor");
        Serial.println("Press - or z to decrease calibration factor");
        scale.set_scale();
        scale.tare();  //Reset the scale to 0
        long zero_factor = scale.read_average(); //Get a baseline reading
        Serial.print("Zero factor: ");
        Serial.println(zero_factor);
  myRA.clear(); // explicitly start clean
}
void bukaPakan()
{
        // in steps of 1 degree
        servoPakan.write(45);
        delay(15);                          // waits 15ms for the servo to
reach the position
}
void tutupPakan()
```

```
{
	// in steps of 1 degree
	servoPakan.write(0);
	delay(15);                          // waits 15ms for the servo to
reach the position
}
void loop()
{
	DateTime now = rtc.now();
	y = String (now.year(), DEC);
	m = String (now.month(), DEC);
	dat = String (now.day(), DEC);
	dy = daysOfTheWeek[now.dayOfTheWeek()];
	jam = String (now.hour(), DEC);
	ment = String (now.minute(), DEC);
	detk = String (now.second(), DEC);
	ertc = y + "/" + m + "/" + dat + " (" + dy + ") " + jam +":" +
ment + ":" + detk;
	Serial.println(ertc);
	delay(5000);
	int waktu;
	waktu = now.second();
	//loadcell
	scale.set_scale(calibration_factor);
	//==========Proses Sensing Berat=================
	units = scale.get_units(), 10;
	if (units < 0)
	{
		units = 0.00;
	}
	ounces = units * 0.035274;
  myRA.addValue(units);
 int beratAddRata = myRA.getAverage();
  Serial.println(beratAddRata); //using average
//	Serial.print(units); //not using
	Serial.print(" grams");
	Serial.print("calibration_factor: ");
	Serial.print(calibration_factor);
	Serial.println();
	if(Serial.available())
	{
		char temp = Serial.read();
```

```
        if(temp == '+' || temp == 'a')
        {
                calibration_factor += 1;
        }

        else if(temp == '-' || temp == 'z')
        {
                calibration_factor -= 1;
        }
}
// ==========Proses Sensing Suhu=================
sensors.requestTemperatures();
Serial.print ("suhu : ");
Serial.println(sensors.getTempCByIndex(0));
float tempC = sensors.getTempCByIndex(0);
// =========Proses pengiriman ke esp============
temp = tempC;
d = String(temp,DEC);
String f = String(beratAddRata,DEC);
String swaktu = String(waktu,DEC);
String kirim = d + "#"+swaktu+"#"+f;
e = kirim.c_str();
Serial.println(waktu);
if (waktu >=25 && waktu <=30)
{
        if (count >= 0)
        {
                count= count + 1;
                if (bukaP == true)
                {
                        serial1.write(e);
                        bukaPakan();
                        delay(3000);
                        bukaP = false;
                }
                if(bukaP==false)
                {
                        bukaP = true;
                        serial1.write(e);
                        tutupPakan();
                        Serial.println("tutup pakan");
```

```
                    }
                }
        }
        else
        {
                serial1.write(e);
        }
}
```

Lampiran 3 *Source code login* di android

Code

```
package com.example.saifud.coba;
import android.content.Intent;
import android.os.Bundle;
import android.os.PersistableBundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class LoginActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.login_layout);
        final EditText uname = (EditText) findViewById(R.id.username);
final EditText passwd = (EditText) findViewById(R.id.password);
        final Button submit = (Button) findViewById(R.id.submit);
        submit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
Log.w("Debug", uname.getText().toString());
                Log.w("Debug", passwd.getText().toString());
                if((uname.getText().toString().equals("saifud")) &&
(passwd.getText().toString().equals("12345")) ){
                    Log.w("Sudah Masuk", uname.getText().toString());
                    Toast.makeText(getApplicationContext(), "Login
Berhasil!",Toast.LENGTH_LONG).show();
                    Intent intent = new Intent(LoginActivity.this,
MainActivity.class);
                    startActivity(intent);
                }
                else{
                    Toast.makeText(getApplicationContext(), "Login
Gagal!",Toast.LENGTH_LONG).show();
                }
            }
        });
    }
}
```

Lampiran 4 *Source code* MainAktivity di android

Code

```
package com.example.saifud.coba;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.widget.TextView;
import org.eclipse.paho.client.mqttv3.IMqttDeliveryToken;
import org.eclipse.paho.client.mqttv3.MqttCallback;
import org.eclipse.paho.client.mqttv3.MqttCallbackExtended;
import org.eclipse.paho.client.mqttv3.MqttMessage;
import org.eclipse.paho.client.mqttv3.MqttTopic;
import handler.MqttHandler;
public class MainActivity extends AppCompatActivity {
    MqttHandler mqttHandler;
    TextView suhu;
    TextView pakan;
    TextView berat;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        suhu= (TextView) findViewById(R.id.suhu);
        pakan = (TextView) findViewById(R.id.pakan);
        berat = (TextView) findViewById(R.id.berat);
        startMqtt();
    }
    private void startMqtt(){
        Log.w("coba","saifud test agai");
        mqttHandler= new MqttHandler(getApplicationContext());
        mqttHandler.mqttAndroidClient.setCallback(new
MqttCallbackExtended() {
            @Override
            public void connectComplete(boolean reconnect, String
serverURI) {
                Log.w("Debug",serverURI);
            }

            @Override
            public void connectionLost(Throwable cause) {

            }

            @Override
            public void messageArrived(String topic, MqttMessage
message) throws Exception { //pesan ditrima
                Log.w("Debug", topic.toString());
                Log.w("Debug",message.toString());
                if (topic.toString().equals("saifud/suhu")){
                    suhu.setText(message.toString()+ " °C ");
                }
                else if(topic.toString().equals("saifud/pakan")){
                    pakan.setText(message.toString());
                }
                else if(topic.toString().equals("saifud/berat")){
                    berat.setText(message.toString()+ " gram ");
                }
            }

            @Override
            public void deliveryComplete(IMqttDeliveryToken token) {

            }        });}}
```

Lampiran 5 *Source code* MqttHantdler di android

Code

```
package handler;
        import android.content.Context;
        import android.util.Log;
        import org.eclipse.paho.android.service.MqttAndroidClient;
        import
org.eclipse.paho.client.mqttv3.DisconnectedBufferOptions;
        import org.eclipse.paho.client.mqttv3.IMqttActionListener;
        import org.eclipse.paho.client.mqttv3.IMqttDeliveryToken;
        import org.eclipse.paho.client.mqttv3.IMqttToken;
        import org.eclipse.paho.client.mqttv3.MqttCallbackExtended;
        import org.eclipse.paho.client.mqttv3.MqttConnectOptions;
        import org.eclipse.paho.client.mqttv3.MqttException;
        import org.eclipse.paho.client.mqttv3.MqttMessage;
public class MqttHandler {
    public MqttAndroidClient mqttAndroidClient;
    final String serverUri = "tcp://192.168.43.229:1883"; //deklarasai
    final String clientId = "Blabla";
    final String subscriptionTopicSuhu = "saifud/+";
    final String subscriptionTopicPakan = "saifud/pakan";
    public MqttHandler(Context context){
        mqttAndroidClient = new MqttAndroidClient(context, serverUri,
clientId);
        mqttAndroidClient.setCallback(new MqttCallbackExtended() {
            @Override
            public void connectComplete(boolean b, String s) {
                Log.w("mqtt", s);
            }
            @Override
            public void connectionLost(Throwable throwable) {
                Log.d("Lost", "Connection lost");
            }
            @Override
            public void messageArrived(String topic, MqttMessage
mqttMessage) throws Exception {// penerimaan data
                Log.w("Mqtt", mqttMessage.toString());
            }
            @Override
            public void deliveryComplete(IMqttDeliveryToken
iMqttDeliveryToken) {
            }
        });
        connect();
    }
    private void connect(){
        MqttConnectOptions mqttConnectOptions = new
MqttConnectOptions(); //buat rikonek
        mqttConnectOptions.setAutomaticReconnect(true);
        mqttConnectOptions.setCleanSession(false);
        try {
            mqttAndroidClient.connect(mqttConnectOptions, null, new
IMqttActionListener() {
                @Override
                public void onSuccess(IMqttToken asyncActionToken) {
                    DisconnectedBufferOptions disconnectedBufferOptions
= new DisconnectedBufferOptions();
                    disconnectedBufferOptions.setBufferEnabled(true);
                    disconnectedBufferOptions.setBufferSize(100);
                    disconnectedBufferOptions.setPersistBuffer(false);
disconnectedBufferOptions.setDeleteOldestMessages(false);
mqttAndroidClient.setBufferOpts(disconnectedBufferOptions);
                    subscribeToTopic();
```

```
                }
                @Override
                public void onFailure(IMqttToken asyncActionToken,
Throwable exception) {
                    Log.w("Gagal", "Failed to connect to: " + serverUri
+ " " + exception.toString());
                }
            });
        } catch (MqttException ex){ //jika error manggil mqttexception
            ex.printStackTrace();
        }
    }
    private void subscribeToTopic() {
        try {
            mqttAndroidClient.subscribe(subscriptionTopicSuhu, 0, null,
new IMqttActionListener() {
                @Override
                public void onSuccess(IMqttToken asyncActionToken) {
                    Log.w("Nyambung","Subscribed!");
                }

                @Override
                public void onFailure(IMqttToken asyncActionToken,
Throwable exception) {
                    Log.w("Gagal", "Subscribed fail!");
                }
            });

        } catch (MqttException ex) {
            System.err.println("Exception whilst subscribing");
            ex.printStackTrace();
        }
    }
}
```